

刻录系统盘

打开[UltraSO](#)，选择目录栏中的：文件 > 打开，选中iso文件； 启动 > 写入硬盘镜像；

弹窗后首先主要注意的是：**硬盘驱动器**是否为需要写入的U盘；写入方式为：**USB-HDD+**；最后点击下面的写入，一般都会弹窗提醒需要**格式化U盘**，确认即可；

如果不是USB-HDD+，可能会导致电脑无法通过U盘正常启动；

替换系统源及安装必要ssh-server、git、vim

[Ubuntu 镜像使用帮助](#)

```
#!/bin/bash

sudo add-apt-repository ppa:jonathonf/vim

sudo apt update --yes

sudo apt install -y git

sudo apt install vim

# install openssh-server for remote login
sudo apt install -y openssh-server

# sudo systemctl status ssh
```

系统环境

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 18.04.2 LTS
Release:      18.04
Codename:     bionic
$ uname -a
Linux dislab 4.15.0-46-generic #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 2019
x86_64 x86_64 x86_64 GNU/Linux
$ lspci | grep VGA
17:00.0 VGA compatible controller: NVIDIA Corporation GP104 [GeForce GTX 1070
Ti] (rev a1)
65:00.0 VGA compatible controller: NVIDIA Corporation GP104 [GeForce GTX 1070
Ti] (rev a1)
```

安装显卡驱动

1. 查看当前是否已安装显卡驱动, 以及驱动型号: `nvidia-smi` 或者 `dpkg -l | grep -E "nvidia-[0-9]{3}"` 或者 `modinfo nvidia | grep version`.
2. 卸载当前显卡驱动: `sudo apt-get purge nvidia-*`.
3. 查看系统推荐驱动版本: `ubuntu-drivers devices`.
4. 安装显卡驱动:

```
$ sudo add-apt-repository ppa:graphics-drivers/ppa
$ sudo apt-get update
$ sudo apt-get install nvidia-driver-390
```

5. 安装好之后 `reboot`, 使用 `nvidia-smi` 即可查看驱动是否安装成功以及GPU运行状况;

安装(下载)依赖库、软件包

1. 安装依赖库: `sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev`, 其中有一个库系统依赖找不到, 删除即可;
2. 降低GCC、G++版本:

```
$ sudo apt-get install gcc-5
$ sudo apt-get install g++-5
```

替换之前的版本:

```
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-5 50
$ sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-5 50
```

安装CUDA9.0 runfile:

官网链接: [runfile\(local\)](#), 除了Base Installer外, 还有4个Patch.

百度网盘链接: [深度学习服务器配置 Ubuntu 18.04](#), 密码: `w214`.

P.S. 使用 `runfile` 安装有点麻烦, 主要是安装得运行四个 `run` 文件, 同时每个文件还得输入 `accept` 等信息。

下载完成后依次运行:

```

$ sudo sh cuda_9.0.176_384.81_linux.run
$ sudo sh cuda_9.0.176.1_linux.run
$ sudo sh cuda_9.0.176.2_linux.run
$ sudo sh cuda_9.0.176.3_linux.run
$ sudo sh cuda_9.0.176.4_linux.run
$
$ cat >> ~/.bashrc << EOF
> export PATH=/usr/local/cuda-9.0/bin${PATH:+:${PATH}}
> export LD_LIBRARY_PATH=/usr/local/cuda-
9.0/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
> export CUDA_HOME=/usr/local/cuda
> EOF
$ source ~/.bashrc
$ sudo reboot

```

重启后，测试CUDA是否成功：

```

$ cd ~/NVIDIA_CUDA-9.0_Samples/1_Uutilities/deviceQuery
$ make -j4
$ sudo ./deviceQuery
...
Result = PASS
$
$ cat /usr/local/cuda/version.txt # 查看CUDA版本
CUDA Version 9.0.176
CUDA Patch Version 9.0.176.1
CUDA Patch Version 9.0.176.2
CUDA Patch Version 9.0.176.3
CUDA Patch Version 9.0.176.4

```

下载安装cudnn-9.0-linux-x64-v7.4.1.5.tgz

[官网链接\(需登录才能下载\)](#), 百度网盘已经给出；

```

$ tar -zxvf cudnn-9.0-linux-x64-v7.4.1.5.tgz
$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include
$ sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h
/usr/local/cuda/lib64/libcudnn*

```

查看cudnn版本：

```
$ cat /usr/local/cuda/include/cudnn.h | grep CUDNN_MAJOR -A 2
#define CUDNN_MAJOR 7
#define CUDNN_MINOR 4
#define CUDNN_PATCHLEVEL 1
--
#define CUDNN_VERSION (CUDNN_MAJOR * 1000 + CUDNN_MINOR * 100 +
CUDNN_PATCHLEVEL)

#include "driver_types.h"
```

安装miniconda

将下面代码写入bash文件，如 `miniconda-install.sh`，`sudo bash xxx.sh` 运行即可；

```
#!/bin/bash

sudo apt update
sudo apt upgrade

cd ~

wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O
~/miniconda.sh

sudo chmod +x ~/miniconda.sh

./miniconda.sh && source ~/.bashrc

rm ~/miniconda.sh
```

更改 `pip` 和 `conda` 的源：

```
$ pip install pip -U
$ pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
$ conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsg/free/
$ conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsg/main/
$ conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
$ conda config --set show_channel_urls yes
```

清华大学开源软件镜像站 [Anaconda](https://mirrors.tuna.tsinghua.edu.cn/)、[pypi](https://pypi.tuna.tsinghua.edu.cn/)。

```
$ pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

安装TensorFlow-gpu、PyTorch

1. 在conda环境中安装Tensorflow-1.12、PyTorch

这里使用 `pip` 安装，而不用 `conda` 安装，因为conda安装tensorflow-gpu、以及pytorch(cudatoolkit=9.0)又会下载相应的cuda、cudnn软件包；

```
$ conda create -n tf python=3.6
$ conda activate tf
(tf) $ pip install --upgrade tensorflow-gpu
(tf) $ python
>>> import tensorflow as tf
>>> sess = \
tf.Session(config=tf.ConfigProto(log_device_placement=True))
...
```

注：如果在 `import tensorflow` 时出现: `ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory`，需要配置runtime library:

```
$ sudo bash -c "echo /usr/local/cuda/lib64/ > /etc/ld.so.conf.d/cuda.conf"
$ sudo ldconfig
```

参考: [CSDN libcublas.so.9.0 ImportError](#), [Medium·Install CUDA 9.0 and cuDNN 7.0 for TensorFlow/PyTorch \(GPU\) on Ubuntu 16.04](#)

安装pytorch并测试是否可用cuda:

```
$ conda create -n torch python=3.6
(torch) $ pip install torch torchvision
(torch) $ python
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.cuda.device_count()
2
```

安装jupyter并设置可远程登录

```
(torch) $ pip install ipykernel
# 往jupyter kernel中加入tf环境
(torch) $ python -m ipykernel install --user --name tf
(torch) $ pip install ipython
(torch) $ jupyter notebook --generate-config
~/.jupyter/jupyter_notebook_config.py
(torch) $ ipython
In [1]: from notebook.auth import passwd
In [2]: passwd()
Enter password:
Verfiy password:
Out[2]: 'sha1:b8235f...:02b962484...'
$ cat >> ~/.jupyter/jupyter_notebook_config.py << EOF
```

```
> c.NotebookApp.ip='*'
> c.NotebookApp.password = u'sha1:b8235f...:02b962484...'
> c.NotebookApp.open_browser = False
> c.NotebookApp.allow_remote_access = True
> c.NotebookApp.port =8888
> EOF
```

在 `torch` 环境下执行同样的命令，即可在 `jupyter kernel` 中看到新加入的环境；

参考资料

- [知乎·Ubuntu18.04深度学习环境配置\(CUDA9+cuDNN7.4+TensorFlow1.8\)](#)
- CUDA版本与显卡驱动版本对应关系：[NVIDIA CUDA Toolkit Release Notes](#)
- [从源代码构建](#): 包括tensorflow版本信息及其对应的CUDA、cuDNN版本型号；
- [Medium·Install CUDA 9.0 and cuDNN 7.0 for TensorFlow/PyTorch \(GPU\) on Ubuntu 16.04](#)
- [Medium· Install Tensorflow, PyTorch in Ubuntu 18.04 LTS with Cuda 9.0 for NVIDIA 1080 Ti](#)